Amendments to the Claims:

Please note the insertion of terms "t_allow", "t_no_filter", "t_prevent", "t_open", "t_DD", "t_bad_from", "t_suspect_domain", "t_echo_domain", "t_to_from", "t_forged_domain", "sender_address" and "DD_0" have underscores which may not be visible with the underlining.

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims: We claim:

1) (Currently Amended) An unsolicited message rejecting communications processor connected to

message transfer agents

MTA_0 with an Internet address of IP_0, ~~from address~~ sender_address A_0, declared domain of D_0, and ~~actual~~ real domain of DD_0, and

MTA_1 with an Internet address of IP_1 and ~~to address~~ recipient A_1

comprising:

a) monitoring means for monitoring the communications between MTA_0 and MTA_1;

b) determining means for determining if the communications contains an unsolicited message; and

c) intercepting means for intercepting a RCPT command from MTA_0 and sending an error reply to MTA_0 if the message is determined to be unsolicited[[.]].

whereby MTA_1 controls the interaction between MTA_0 and MTA_1 before a RCPT command from MTA_0 is received by the unsolicited message rejecting communications processor and

whereby the connection with MTA_0 is rejected by the intercepting means before the data portion of the unsolicited message is transmitted.

2) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes [[a]] an allow_address database and wherein the determining means determines if a message is not unsolicited by checking if the IP_0 is in the allow_address database.

3) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes a prevent_address database and wherein the determining means determines if a message is unsolicited by checking if IP_0 is in the prevent_address database.

4) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes access to a open relay database and wherein the determining means determines if a message is unsolicited by checking if IP_0 is in the open relay database.

5) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes access to a DNS (domain name server) database and wherein the determining means determines if a message is unsolicited by checking if IP_0 has a domain name entry DD_0 in the DNS database.

6) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes a bad_from database and wherein the

determining means determines if a message is unsolicited by checking if the ~~from-address~~ sender_address A_0 is in the bad_from database.

7) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes a suspect_domain database and wherein the determining means determines if a message is unsolicited by checking if the ~~actual~~ real domain DD_0 matches the domain of ~~from-address~~ sender_address A_0 and the domain of ~~from-address~~ sender_address A_0 is in the suspect_domain database.

8) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the ~~from-address~~ sender_address A_0 matches the ~~to-address~~ recipient (A_1).

9) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes a no_filter database and wherein the determining means determines if the message is to be blocked if it is determined to be unsolicited by checking if the recipient A_1 is in the no_filter database.

10) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D_0 of MTA_0 is the same as the domain D_1 of MTA_1.

11) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D_0 of MTA_0 does not match the

real domain ~~DD_1~~ DD_0 and the declared domain D_0 is in the suspect_domain database.

12) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes a rejected_connection database which logs the time, ~~from address~~ sender_address A_0, ~~to address~~ recipient A_1, and the reason for the rejection if a message is rejected if the message is determined to be unsolicited.

13) (Currently Amended) The unsolicited message ~~blocking~~ rejecting communications processor in Claim 1, further includes [[a]] an allowed_connection database which logs the time and ~~to address~~ recipient A_1 if the message is determine not to be unsolicited.

14) (Currently Amended) A method for

a receiving networked computer system with an Internet connection, a mail transport agent MTA_1, an Internet address IP_1, ~~to-address~~ recipient A_1, and an operating system capable of executing the method

to reject unsolicited messages from

a transmitting networked computer system with an Internet connection and a message transfer agent MTA_0, an Internet address IP_0, ~~from-address~~ sender_address A_0, declared domain D_0, and ~~actual~~ real domain DD_0

comprising the steps of:

a) waiting for a new SMTP connection request;

b) relaying and monitoring the replies from MTA_0 to MTA_1;

c) relaying replies from MTA_1 to MTA_0;

c') allowing MTA_1 to control the interaction between MTA_0 and MTA_1 until a RCPT reply is received from MTA_0;

d) intercepting the RCPT reply from MTA_0 to MTA_1;

e) determining if the message is unsolicited by analyzing the monitored replies;

f) releasing the intercepted RCPT reply if the message is determined not to be unsolicited; ~~and~~

g) sending [[a]] an error reply to MTA_0 if the message is determined to be unsolicited; and

h) rejecting the connection with MTA_0 before the data portion of the unsolicited message is transmitted if the message is determined to be unsolicited.

~~whereby MTA_1 controls the interaction between MTA_0 and MTA_1 until a RCPT command is received from MTA_0 and~~

~~whereby the connection with MTA_0 is rejected before the data portion of the unsolicited message is transmitted.~~

15) (Currently Amended) A method for

a receiving networked computer system with an Internet connection, a mail transport agent MTA_1, IP address IP_1, a domain name D_1, a recipient, A_1, an allow_address database, a prevent_address database, a suspect_domain database, a bad_from database, a no_filter database, a rejected_connection database, an allowed_connection database, and an operating system capable of executing the method

to reject unsolicited messages from

a transmitting networked computer system with an Internet connection, a message transfer agent MTA_0, an IP address of IP_0, a declared domain name D_0, a real domain name DD_0, and a sender address of A_0

comprising the steps of:

a) waiting for a SMTP connection request on the receiving networked computer system's Internet connection;

b) sending a 220 reply to MTA_0 to acknowledge the requested connection;

c) extracting IP address IP_0 from the connection request;

d) testing if the DNS database has a domain name DD_0 for IP_0;

e) testing if IP_0 is in an open relay database;

f) testing if IP_0 is in the allow_address database;

g) testing if IP_0 is in the prevent_address database[[,]];

h) requesting a connection with MTA_1;

i) waiting for a 220 reply from MTA_1 to acknowledge the requested connection;

j) waiting for a reply from either MTA_0 or MTA_1;

k) jumping to step n) if the reply is not from MTA_1;

l) relaying the reply from MTA_1 to MTA_0;

m) jumping to step j) to wait for a new reply;

n) jumping to step t) if the reply from MTA_0 is not a **HELO**;

o) extracting domain D_0 from the reply;

p) testing if declared domain D_0 of MTA_0 matches domain D_1 of MTA_1;

q) testing if declared domain D_0 does not match real domain DD_0 of MTA_0 AND declared domain D_0 is in the suspect_domain database;

r) relaying the HELO reply from MTA_0 to MTA_1;

s) jumping to step j) to wait for a new reply;

t) jumping to step z) if reply from MTA_0 is not a **MAIL**;

u) extracting ~~from-address~~ sender_address A_0;

v) testing if A_0 is in the bad_from database;

w) testing if DD_0 does not match the domain of A_0 and the domain of A_0 is in the suspect_domain database;

x) relaying MAIL reply to MTA_1;

y) jumping to step j) to wait for a new reply;

z) jumping to step kk) if reply from MTA_0 is not a **RCPT**;

aa) extracting ~~to-address~~ recipient A_1;

bb) testing if A_1 is in no_filter database;

cc) testing if A_0 matches A_1;

dd) jumping to step hh) if NOT(t_allow OR t_no_filter OR NOT (t_prevent OR t_open OR t_DD[[-]]) OR t_bad_from OR t_suspect_domain OR t_to_from OR t_echo_domain OR t_forged_domain)[[)]] ;

ee) logging time and ~~to-address~~ recipient A_1 in the allowed_connection database;

ff) relaying RCPT reply to MTA_1;

gg) jumping to step j) to wait for a new reply;

hh) logging the time, from-address sender_address A_0, ~~to-address~~ recipient A_1, and the reason for rejecting the connection in the rejected_connection database;

ii) rejecting the connection to MTA_0 by sending a 550 reply to MTA_0;

jj) jumping to step j) to wait for a new reply;

kk) jumping to step vv) if reply from MTA_0 is not **DATA**;

ll) relaying DATA reply to MTA_1;

mm) waiting for a 354 reply from MTA_1;

nn) relaying the 354 reply from MTA_1 to MTA_0;

oo) waiting for the data from MTA_0;

pp) relaying the data from MTA_0 to MTA_1;

qq) waiting for a .\r\n from MTA_0;

rr) relaying the .\r\n from MTA_0 to MTA_1;

ss) waiting for a 250 reply from MTA_1;

tt) relaying the 250 reply to MTA_0;

uu) jumping to step j) to wait for a new reply;

vv) jumping to step yy) if reply from MTA_0 is not **RSET, SEND, ~~SCML~~ SOML, SAML, VRFY, NOOP, EXPN, HELP,** or **TURN;**

ww) relaying reply to MTA_1;

xx) jumping to step j) to wait for a new reply;

yy) jumping to step ddd) if reply from MTA_0 is not a **QUIT**;

zz) relaying the QUIT reply to MTA_1;

aaa) waiting for 221 reply from MTA_1;

bbb) relaying 221 reply from MTA_1 to MTA_0;

ccc) jumping to step a) to wait for a new connection;

ddd) sending a 500 reply to MTA_0 to signal a syntax error; and

eee) jumping to step a) to wait for a new connection[[.]],

wherein t_allow represents the results of the testing in step (f); t_no_filter represents the results of the testing in step (bb); t_prevent represents the results of the testing in step (g); t_open represents the results of the testing in step (e); t_DD represents the results of the testing in step (d); t_bad_from represents the results of the testing in step (v); t_suspect_domain represents the results of the testing in step (w); t_echo_domain represents the results of the testing in step (p); t_to_from represents the results of the testing in step (cc); and t_forged_domain represents the results of the testing in step (q).

16) (New) The method of claim 14, wherein the determining comprises checking if the IP_0 is in a allow_address database.

17) (New) The method of claim 14, wherein the determining comprises checking if IP_0 is in a prevent_address database.

18) (New) The method of claim 14, wherein the determining comprises checking if IP_0 has a domain name entry DD_0 in a DNS database.

19) (New) The method of claim 14, wherein the determining comprises checking if the real domain DD_0 matches the domain of sender_address A_0 and the domain of sender_address A_0 is in a suspect_domain database.

20) (New) The method of claim 14, wherein the determining comprises checking if the declared domain D_0 of MTA_0 does not match the real domain DD_0 and the declared domain D_0 is in the suspect_domain database.